Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Regret Ratio Minimization in Multi-Objective Submodular Function Maximization

Tasuku Soma
The University of Tokyo
tasuku_soma@mist.i.u-tokyo.ac.jp

Yuichi Yoshida
National Institute of Informatics, *and*
Preferred Infrastructure, Inc.
yyoshida@nii.ac.jp

Abstract

Submodular function maximization has numerous applications in machine learning and artificial intelligence. Many real applications require multiple submodular objective functions to be maximized, and which function is regarded as important by a user is not known in advance. In such cases, it is desirable to have a small family of representative solutions that would satisfy any user's preference. A traditional approach for solving such a problem is to enumerate the Pareto optimal solutions. However, owing to the massive number of Pareto optimal solutions (possibly exponentially many), it is difficult for a user to select a solution. In this paper, we propose two efficient methods for finding a small family of representative solutions, based on the notion of regret ratio. The first method outputs a family of fixed size with a non-trivial regret ratio. The second method enables us to choose the size of the output family, and in the biobjective case, it has a provable trade-off between the size and the regret ratio. Using real and synthetic data, we empirically demonstrate that our methods achieve a small regret ratio.

Introduction

Submodular function maximization has numerous applications in machine learning and artificial intelligence, such as budget allocation (Soma et al. 2014), document summarization (Lin and Bilmes 2010; 2011), maximum entropy sampling (Ko, Lee, and Queyranne 1995), online service privacy (Krause and Horvitz 2008), and sensor placement (Krause, Singh, and Guestrin 2008). Many efficient algorithms have been developed to solve these problems by maximizing a single submodular function.

However, in real applications, we often face multiple conflicting criteria. For example, in data summarization, we are to select a subset of a data set that maximizes two criteria: coverage and diversity. That is, we are to find a subset that explains the entire data well, and at the same time, elements in the subset are different to each other. Further, in the budget allocation problem, we are to buy ads to maximize the expected number of people influenced by ads, while we also need to minimize the cost of buying ads. These problems prompt us to consider maximizing *multiple* submodular functions.

In contrast to maximizing a single submodular function, maximizing multiple submodular functions is not well understood. The difficulty in multi-objective optimization arises from the fact that there may be no single solution that maximizes all the objective functions simultaneously. Hence, preferable solutions can vary from one user to another, depending on which objective function is more important to the user. Moreover, a user often cannot describe his/her own preference explicitly but can only compare two solutions based on his/her implicit preference. In such cases, a natural goal is to precompute a family of "representative" solutions so that a user with any preference can find an (almost) optimal set in the family.

A standard approach for finding such a family is to enumerate the Pareto optimal solutions. However, this approach has two drawbacks: (i) The number of Pareto optimal solutions is often massive, and enumerating all of them does not enable a user to select a solution. (ii) No efficient algorithm for computing the Pareto optimal solutions is known when the objective functions are submodular.

Our contributions

In this paper, we tackle the above-mentioned problem using the concept of *regret ratio*, introduced in (Nanongkai et al. 2010). Here, we assume that the preference of a user can be expressed as a convex combination of the objective functions. Then, intuitively speaking, the regret ratio of a family of solutions is the (normalized) loss caused by choosing a solution from the family instead of considering all feasible solutions. The advantage of introducing such a concept and optimizing it is that we can control the size of the family.

In this paper, we formalize the concept of regret ratio for multi-objective submodular function maximization. Then, to find a family of solutions with a small regret ratio, we propose two methods, namely the coordinate-wise maximum method and the polytope method. The coordinate-wise maximum method outputs a family of fixed size with a non-trivial regret ratio. The polytope method enables us to choose the size of the output family, and in the biobjective case, it has a provable trade-off between the size and the regret ratio. Both methods can handle monotone and non-monotone submodular functions under any constraint as long as there is an approximation algorithm for the corresponding problem on a *single* submodular function. In addition, we show that the trade-off attained by the polytope method cannot be improved significantly by presenting a lower bound instance. Finally, we experimentally demonstrate the superiority of our methods on a data summarization problem and the budget allocation problem.

Related Work

The notion of regret ratio was originally introduced for obtaining a subset of representative points from a point set (Nanongkai et al. 2010). Several notions of representative sets have been proposed, including *k-representative skyline queries* (Lin et al. 2007; Tao et al. 2009), *top-k dominating queries* (Yiu and Mamoulis 2009), and *ϵ-skyline queries* (Xia, Zhang, and Tao 2008). In comparison to these notions, regret ratio has the following desirable properties: (i) *scale invariance*, i.e., even if we multiply the values of some coordinate by a positive constant, the regret ratio of a set remains unchanged; (ii) *stability*, i.e., adding a point that is unimportant, in the sense that it is not optimal for any preference, does not change the regret ratio of a set; (iii) *parameter-freeness*, i.e., only the number of points to be selected is required. These features strongly motivate us to compute a family of solutions with a small regret ratio in the multi-criteria setting.

We note that for point sets, there is an algorithm with a provable trade-off between the size of the output set and its regret ratio (Nanongkai et al. 2010). However, this algorithm cannot be directly applied to our submodular setting because it checks all the points, which takes exponential time in our setting.

In a similar problem, the robust submodular function maximization problem (Krause et al. 2008), multiple monotone submodular functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$ and an integer k are given; the goal is to find a set $S \subseteq E$ of size at most k that maximizes $\min\{f_1(S), \dots, f_d(S)\}$. In our problem, we consider a (unknown) convex combination of $f_1, \dots, f_d$ and output a family of sets instead of a single set.

The linear submodular bandit problem (Yue and Guestrin 2011; Krause, Roper, and Golovin 2011) also considers convex combinations of submodular objective functions. In this problem, convex coefficients are drawn from some unknown distribution and one can learn the distribution with sampling and optimization. On the other hand, our setting is *adversarial* in the sense that we must consider all possible convex combinations.

Preliminaries

For an integer k, let $[k]$ denote the set $\{1, 2, \dots, k\}$. We denote the set of nonnegative reals by $\mathbb{R}_+$. Let E be a finite ground set. A function $f : 2^E \to \mathbb{R}$ is said to be *submodular* if

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$$

for every $X, Y \subseteq E$. It is well known that submodularity is equivalent to the *diminishing return property*: $f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y)$ for every $X \subseteq Y \subsetneq E$ and $e \in E \setminus Y$.

For functions $f_1, \dots, f_d : 2^E \to \mathbb{R}$ and a vector $\boldsymbol{a} \in \mathbb{R}^d$, we define a function $f_{\boldsymbol{a}}(X) := \sum_{i=1}^{d} a(i) f_i(X)$. Note that, if $f_1, \dots, f_d$ are submodular and $\boldsymbol{a} \in \mathbb{R}^d_+$, then $f_{\boldsymbol{a}}$ is also submodular.

Regret-minimizing family

Let $\mathcal{C} \subseteq 2^E$ be a family of sets, which we regard as a constraint on solutions. Let $\mathcal{S} \subseteq \mathcal{C}$ be a subfamily of $\mathcal{C}$ and $f : 2^E \to \mathbb{R}_+$ be a function. We define the *regret* of $\mathcal{S}$ with respect to f under the constraint $\mathcal{C}$ as $r_{f,\mathcal{C}}(\mathcal{S}) := \max_{X \in \mathcal{C}} f(X) - \max_{X \in \mathcal{S}} f(X)$. Then, we define the regret ratio of $\mathcal{S}$ with respect to f under $\mathcal{C}$ as

$$\mathrm{rr}_{f,\mathcal{C}}(\mathcal{S}) = \frac{r_{f,\mathcal{C}}(\mathcal{S})}{\max_{X \in \mathcal{C}} f(X)} = 1 - \frac{\max_{X \in \mathcal{S}} f(X)}{\max_{X \in \mathcal{C}} f(X)}.$$

Note that $\mathrm{rr}_{f,\mathcal{C}} \in [0, 1]$ and that $\mathrm{rr}_{f,\mathcal{C}}(\mathcal{S})$ represents the normalized loss caused by choosing a solution from $\mathcal{S}$ instead of $\mathcal{C}$. Then, the *(maximum) regret ratio* of $\mathcal{S}$ with respect to functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$ is defined as

$$\mathrm{rr}_{f_1,\dots,f_d,\mathcal{C}}(\mathcal{S}) = \max_{\boldsymbol{a} \in \mathbb{R}^d_+} \mathrm{rr}_{f_{\boldsymbol{a}},\mathcal{C}}(\mathcal{S}).$$

Intuitively speaking, $\boldsymbol{a} \in \mathbb{R}^d_+$ represents a preference of a user on the functions $f_1, \dots, f_d$, and $\mathrm{rr}_{f_1,\dots,f_d,\mathcal{C}}(\mathcal{S})$ is the worst regret ratio over all the preferences. We often omit the subscripts of $f_1, \dots, f_d$ when they are clear from the context.

We study the following problem in this paper:

Definition 1 (Regret ratio minimization in multi-objective submodular function maximization)**.** Given submodular functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$, $\mathcal{C} \subseteq 2^E$, and $k \in \mathbb{N}$, find $\mathcal{S} \subseteq \mathcal{C}$ with $|\mathcal{S}| \leq k$ that minimizes the maximum regret ratio $\mathrm{rr}_{f_1,\dots,f_d,\mathcal{C}}(\mathcal{S})$.

If $\boldsymbol{a}$ is fixed, finding $X^* \in \mathcal{C}$ that maximizes $f_{\boldsymbol{a}}(X)$ is called *submodular function maximization*, which is an NP-hard problem in general. However, for various constraint families $\mathcal{C}$, one can find an approximate solution efficiently. If one can find an α-approximate solution X^*, the corresponding regret ratio is $1 - \frac{f_{\boldsymbol{a}}(X^*)}{\max_{X \in \mathcal{C}} f(X)} \leq 1 - \alpha$.

Geometric Interpretation

The multi-objective submodular function maximization has a nice geometric interpretation. Let us consider a function $\boldsymbol{f}(X) := [f_1(X) \dots f_d(X)]^\top \in \mathbb{R}^d_+$. Note that $f_{\boldsymbol{a}}(X) = \boldsymbol{a}^\top \boldsymbol{f}(X)$. For $\mathcal{S} \subseteq \mathcal{C}$, we define $C_{\boldsymbol{f}}(\mathcal{S}) := \mathrm{conv}\{\boldsymbol{f}(X) : X \in \mathcal{S}\}$. We associate $\mathcal{S} \subseteq \mathcal{C}$ with a polytope

$$P(\mathcal{S}) := \{\boldsymbol{x} \in \mathbb{R}^d_+ : \exists \boldsymbol{y} \in C_{\boldsymbol{f}}(\mathcal{S}) \text{ s.t. } \boldsymbol{x} \leq \boldsymbol{y}\},$$

where $\boldsymbol{x} \leq \boldsymbol{y}$ means $x(i) \leq y(i)$ $(i \in [d])$.

Lemma 2 ((Peng and Wong 2014, Lemma 1))**.** $\mathrm{rr}_{f_1,\dots,f_d,\mathcal{C}}(\mathcal{S}) \leq 1 - \alpha$ *if and only if* $P(\mathcal{C}) \subseteq \alpha^{-1} P(\mathcal{S})$.

The above characterization establishes that the maximum regret ratio is *scale-invariant*, i.e., even if we replace f_i with βf_i for some $\beta > 0$, the minimum regret ratio is preserved. The following lemma is just a restatement of the above lemma, but is useful for the analysis of our algorithms. A *frontier* face is a face of $P(\mathcal{S})$ consisting of Pareto optimal points.

Lemma 3. $\mathrm{rr}_{f_1,\dots,f_d,\mathcal{C}}(\mathcal{S}) = \max_{\boldsymbol{a}} \mathrm{rr}_{f_{\boldsymbol{a}},\mathcal{C}}(\mathcal{S})$, *where* $\boldsymbol{a}$ *runs over the nonnegative normal vectors of all frontier faces of* $P(\mathcal{S})$.

Algorithm 1 Coordinate-wise maximum method

Input: Submodular functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$, a constraint $\mathcal{C} \subseteq 2^E$, and an approximation algorithm $\mathcal{A}_i$ for $\max_{X \in \mathcal{C}} f_i(X)$ $(i \in [d])$.

for $i \in [d]$ **do**
  $X_i \leftarrow$ a solution obtained by applying $\mathcal{A}_i$ to f_i.
return $\mathcal{S}_{\text{coord}} := \{X_1, \dots, X_d\}$.

Algorithms

In this section, we present two algorithms. Both algorithms require approximation algorithms for maximizing submodular functions. Let α be the minimum approximation ratio of these approximation algorithms. The first algorithm, the coordinate-wise maximum method, always outputs a family of d solutions with regret ratio $1 - \alpha/d$. The second algorithm, the polytope method, has a provable guarantee only when $d = 2$. However, it has a trade-off between the regret ratio and the size of the output, and the regret ratio converges to $1 - \alpha$ as the output size increases.

Coordinate-wise maximum method

Besides functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$ and a constraint $\mathcal{C} \subseteq 2^E$, the coordinate-wise maximum method requires an approximation algorithm $\mathcal{A}_i$ for $\max_{X \in \mathcal{C}} f_i(X)$ $(i \in [d])$. Then, it simply computes an approximate solution X_i for $\max_{X \in \mathcal{C}} f_i(X)$ by using $\mathcal{A}_i$ for each $i \in [d]$, and subsequently outputs $\mathcal{S}_{\text{coord}} := \{X_1, \dots, X_d\}$. See Algorithm 1 for further details.

Lemma 4. *Let α be the minimum approximation ratio of $\mathcal{A}_i$'s. Then, we have $\mathrm{rr}_{\mathcal{C}}(\mathcal{S}_{\text{coord}}) \leq 1 - \frac{\alpha}{d}$.*

Proof. For any $\boldsymbol{a} \in \mathbb{R}_+^d$, we have

$$\begin{aligned}
\max_{i \in [d]} f_{\boldsymbol{a}}(X_i) &\geq \frac{1}{d} \sum_{i \in [d]} a(i) f_i(X_i) \\
&\geq \frac{\alpha}{d} \sum_{i \in [d]} a(i) \max_{X \in \mathcal{C}} f_i(X) \\
&\geq \frac{\alpha}{d} \max_{X \in \mathcal{C}} \sum_{i \in [d]} a(i) f_i(X) \\
&= \frac{\alpha}{d} \max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X).
\end{aligned}$$

Therefore, we have

$$\mathrm{rr}_{\mathcal{C}}(\mathcal{S}_{\text{coord}}) = \max_{\boldsymbol{a} \in \mathbb{R}_+^d} \left[1 - \frac{\max_{i \in [d]} f_{\boldsymbol{a}}(X)}{\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)} \right] \leq 1 - \frac{\alpha}{d}. \qquad \square$$

We have the following:

Theorem 5. *Suppose that $\mathcal{A}_i$ is an α-approximation algorithm for $\max_{X \in \mathcal{C}} f_i(X)$ with time complexity $T_i(|E|)$ for $i \in [d]$. Then, Algorithm 1 outputs a family of d solutions with regret ratio at most $1 - \alpha/d$ in $O(d + \sum_{i \in [d]} T_i(|E|))$ time.*

Algorithm 2 Polytope method

Input: Submodular functions $f_1, \dots, f_d : 2^E \to \mathbb{R}_+$, a constraint $\mathcal{C} \subseteq 2^E$, an integer $k \in \mathbb{N}$, and an approximation algorithm $\mathcal{A}$ for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ $(\boldsymbol{a} \in \mathbb{R}_+^d)$.

for $i \in [d]$ **do**
  $X_i \leftarrow$ a solution obtained by applying $\mathcal{A}$ to f_i.
$\mathcal{S} \leftarrow \{X_1, \dots, X_d\}$, $P \leftarrow P(\mathcal{S})$.
while $|\mathcal{S}| < k$ **do**
  for each frontier face F of P **do**
    Find a nonnegative normal vector $\boldsymbol{a}$ of F.
    $X \leftarrow$ a set obtained by applying $\mathcal{A}$ to $f_{\boldsymbol{a}}$.
    Add X to $\mathcal{S}$.
    if $|\mathcal{S}| = k$ **then return** $\mathcal{S}$.
  $P \leftarrow P(\mathcal{S})$.
return $\mathcal{S}$.

Proof. The regret ratio is immediate from Lemma 4. The time complexity follows as we run the algorithm $\mathcal{A}_i$ for $i \in [d]$ and the output set S_{coord} has size d. $\square$

Polytope method

Our second algorithm is based on the geometric characterization of the regret ratio. The algorithm first runs Algorithm 1 to obtain a polytope $P(\mathcal{S})$. For each frontier face F of $P(\mathcal{S})$, we compute a nonnegative normal vector $\boldsymbol{a}$ of F. Note that one can always find a nonnegative normal vector $\boldsymbol{a}$ from the definition of $P(\mathcal{S})$. Then, we run an approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ to obtain an approximate solution X, and add X to $\mathcal{S}$. A pseudocode description is presented in Algorithm 2.

To explain the intuitive concept underlying this algorithm, let us consider the case of $d = 2$. An illustration of the algorithm is shown in Figure 1. In the figure, we identify a solution X with a point $\boldsymbol{f}(X)$. The algorithm tries to reduce the area of the region that may contain points not included by $P(\mathcal{S})$, which is shown as the shaded region in Figure 1. Intuitively, the shaded region can be shrunk by taking a normal vector of the face and adding a point maximizing $f_{\boldsymbol{a}}(X)$.

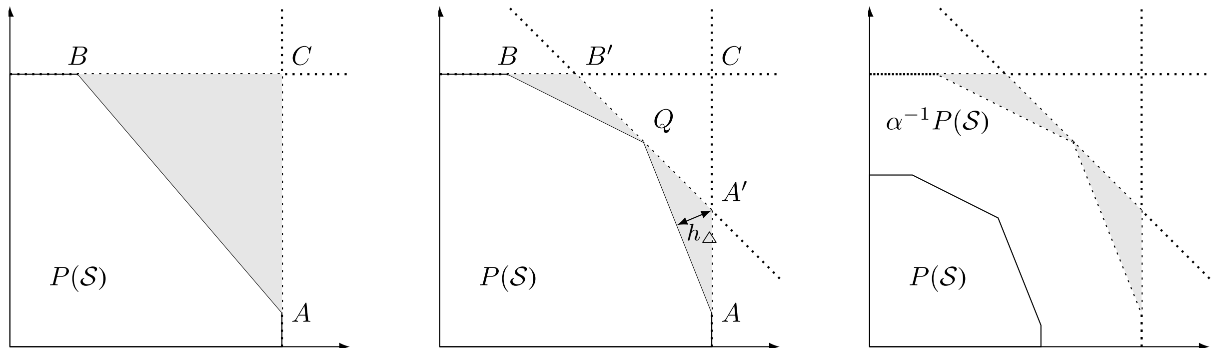
Before analyzing the regret ratio of Algorithm 2, we analyze its time complexity:

Theorem 6. *Suppose $\mathcal{A}$ is an approximation algorithm with time complexity $T(|E|)$. Then, Algorithm 2 runs in $O(k \log k + k^{\lfloor d/2 \rfloor} + (d + k)T(|E|))$ time.*

Proof. Through the algorithm, the number of invocations of $\mathcal{A}$ is $O(d + k)$.

The process of maintaining the faces is essentially equivalent to the dynamic update of a convex hull in d-dimensional space. As we end with adding k points, we can maintain the faces in $O(k \log k + k^{\lfloor d/2 \rfloor})$ time by using the algorithm by (Clarkson and Shor 1989).

Summing up these time complexities, we get the desired result. $\square$



(a) Beginning of algorithm (b) After the first loop (c) Approximate case

Figure 1: Investigation of faces by Polytope Algorithm. The x-axis and y-axis represent the values of f_1 and f_2, respectively, and a solution X is identified with a point $\boldsymbol{f}(X)$. (a) The Coordinate-wise maximum method yields points A and B. Then we know that other Pareto points must be below of C. The initial uncovered region is $\triangle ABC$. (b) Next, the algorithm picks a normal vector of face AB and finds point Q. Then we know that other Pareto points are also below of line $B'A'$, which is a line passing through Q and parallel to face AB. Now the uncovered region shrinks into $\triangle A'AQ$ and $\triangle B'BQ$. (c) For the approximate case, one can run a similar argument, but the definition of the uncovered region is changed.

Analysis for the exact case First, we analyze Algorithm 2 when $d = 2$, and we can find *exact* solutions for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. Indeed, our algorithm is closely related to the *Chord* algorithm for approximating convex curves (see (Daskalakis, Diakonikolas, and Yannakakis 2010) and the references therein).

Theorem 7. *Assume that $d = 2$ and that we can find exact solutions for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. After Algorithm 2 investigates all the faces of P i times, the maximum regret ratio $\mathrm{rr}_{\mathcal{C}}(\mathcal{S})$ is at most $\sqrt{2} \cdot 2^{-i}$.*

For the proof, we analyze the area of the region that may contain points not included by $P(\mathcal{S})$. We refer to this region as the *uncovered region*. For example, in Figure 1, the uncovered regions are represented by the shaded regions. Intuitively, in each iteration, the areas of the uncovered regions shrink. Indeed, the areas shrink exponentially.

Lemma 8 ((Daskalakis, Diakonikolas, and Yannakakis 2010, Lemma 3.11), restated in our context.)**.** *Suppose that Algorithm 2 processes face AB. Let $T = \triangle ABC$ be the part of the uncovered region corresponding to face AB. Denote $Q = \boldsymbol{f}(X)$, where X is a solution found in Line 7. Let $T_1 = \triangle AA'Q$ and $T_2 = \triangle BB'Q$ be the parts of the new covered region corresponding to faces AQ and BQ, respectively. Then, we have $S(T_1) + S(T_2) \le S(T)/4$, where $S(T)$ denotes the area of T.*

Proof of Theorem 7. Since the regret ratio is scale-invariant, we can assume that $\max_{X \in \mathcal{C}} f_1(X) = \max_{X \in \mathcal{C}} f_2(X) = \sqrt{2}$. Then, the distance from the origin to any face of $P(\mathcal{S})$ is at least 1, and the area of the initial uncovered region is at most 1. By Lemma 8, after Algorithm 2 processes all the faces of P, the areas of the uncovered regions shrink by a factor of $1/4$. Let us focus on a single triangle $\triangle$ in the uncovered region, and let $h_\triangle$ be the maximum distance from the face to a point in the uncovered region (see Figure 1b). Since $\triangle$ is an obtuse triangle, we have $S(\triangle) \ge h_\triangle^2/2$. Then, $\max_\triangle h_\triangle^2 \le 2\sum_\triangle S(\triangle) = 2S(\text{uncovered region}) \le 2 \cdot 4^{-i}$. Thus, $\max_\triangle h_\triangle \le \sqrt{2} \cdot 2^{-i}$. By Lemma 3,

$$\begin{aligned}\mathrm{rr}_{\mathcal{C}}(\mathcal{S}) &= \max_\triangle \frac{h_\triangle}{\mathrm{dist}(\triangle, \mathbf{0}) + h_\triangle} \\ &\le \max_\triangle \frac{h_\triangle}{1 + h_\triangle} \\ &\le \max_\triangle h_\triangle \\ &\le \sqrt{2} \cdot 2^{-i},\end{aligned}$$

where the first inequality follows from the fact that the distance from the origin to any triangle is at least 1. ☐

Corollary 9. *After Algorithm 2 adds k solutions to $\mathcal{S}$, $\mathrm{rr}_{\mathcal{C}}(\mathcal{S})$ is at most $\sqrt{2} \cdot 2^{-\lfloor \log_2(k-1) \rfloor} = O(1/k)$.*

Proof. One can check that after Algorithm 2 examines all the faces i times, the number of faces in $P(\mathcal{S})$ is at most $2^i + 1$. Thus, we have $k \le 2^i + 1$, which yields $i \le \lfloor \log_2(k-1) \rfloor$. ☐

Analysis for the approximate case Let us analyze the case where we have only an α-approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. In this case, the best one can hope for is that $P(\mathcal{C}) \subseteq \alpha^{-1} P(\mathcal{S})$, i.e., any Pareto optimal point is within the α-multiplicative factor.

Theorem 10. *Assume that $d = 2$ and that we can find α-approximate solutions for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$. After Algorithm 2 investigates all the faces of P i times, the maximum regret ratio $\mathrm{rr}_{\mathcal{C}}(\mathcal{S})$ is at most $1 - \alpha + \sqrt{2} \cdot 2^{-i}$.*

Proof. The proof idea is showing that $P(\mathcal{C}) \subseteq (\alpha - \epsilon)^{-1} P(\mathcal{S})$, where ϵ decreases exponentially in i. Let us call the area of the region that may contain points not included by $\alpha^{-1} P(\mathcal{S})$ the *uncovered region* (see Figure 1c). It suffices to show the theorem for the case in which the approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ always returns

α-approximate solutions. To see this, suppose that we obtain a β-approximate solution ($\beta > \alpha$) for some normal vector $\boldsymbol{a}$ of a face of $P(\mathcal{S})$. Adding this approximate solution to $\mathcal{S}$ reduces the uncovered area more than adding an α-approximate solution. Thus, the analysis reduces to that of the exact case and the theorem follows from Theorem 7. $\square$

The above argument heavily relies on Lemma 8, which is shown only for the two-dimensional case. In higher dimension, the uncovered region becomes complicated; therefore the analysis becomes more difficult. We leave the analysis in higher dimension for future work.

Having said that, we believe that our analysis of the biobjective case is useful because the biobjective case has many important applications as stated in the introduction.

Lower Bound

In this section, we show that the trade-off achieved by the polytope method (see Corollary 9) cannot be improved significantly even in the two-dimensional case. More specifically, we show the following:

Theorem 11. *For any k, there exist n, $f_1, f_2 : 2^E \to \mathbb{R}_+$ with $|E| = n$, and $\mathcal{C} \subseteq 2^E$ such that an arbitrary subfamily $\mathcal{S} \subseteq \mathcal{C}$ has a maximum regret ratio $\Omega(\frac{1}{k^2})$.*

Proof. Our construction is inspired by (Nanongkai et al. 2010, Theorem 4). Let $f_1(X) := \cos(\frac{\pi|X|}{2n})$ and $f_2(X) := \sin(\frac{\pi|X|}{2n})$. Note that f_1 and f_2 are submodular because $\sin(\frac{\pi x}{2})$ and $\cos(\frac{\pi x}{2})$ are concave for $x \in [0, 1]$. We define $\mathcal{C} := 2^E$, i.e., we do not impose constraints. Let us take an arbitrary $\mathcal{S} \subseteq 2^E$ with $|\mathcal{S}| \leq k$. Without loss of generality, we can assume that two arbitrary distinct elements have different cardinalities (otherwise, we delete some element from $\mathcal{S}$ without losing the regret ratio). We sort the k elements in $\mathcal{S}$ such that $|X_1| < |X_2| < \cdots < |X_k|$. Further, we define $X_0 := \emptyset$ and $X_{k+1} = E$. Let $\phi_i := \frac{\pi|X_i|}{2n}$ ($i = 0, \ldots, k+1$) and define $\theta_i = \phi_i - \phi_{i-1}$ ($i = 1, \ldots, k+1$). Since $\theta_1 + \cdots + \theta_{k+1} = \frac{\pi}{2}$, there exists j such that $\theta_j \geq \frac{\pi}{2(k+1)}$. Define $\beta := \theta_j$. By taking n large enough, we can find $X \subseteq E$ such that $\frac{\pi|X|}{2n} = \phi_j + \frac{\beta}{2} =: \gamma$. Let us consider $\boldsymbol{a} = [\cos\gamma, \sin\gamma]^\top$. One can check that $\max_{X \in 2^E} f_{\boldsymbol{a}}(X) = 1$ and $\max_{X \in \mathcal{S}} f_{\boldsymbol{a}}(X) = f_{\boldsymbol{a}}(X_j) = \cos\gamma\cos\phi_j + \sin\gamma\sin\phi_j = \cos(\frac{\beta}{2})$. Therefore, the regret ratio is $1 - \cos(\frac{\beta}{2}) = \Omega(\frac{\beta^2}{4}) = \Omega(\frac{1}{k^2})$. $\square$

We note that our proof is information theoretic and that it does not rely on any assumption on computational complexity such as P $\neq$ NP.

Experiments

In this section, we experimentally demonstrate that our methods reduce the regret ratio effectively. We conducted experiments on a Linux server with an Intel Xeon E5-2690 (2.90 GHz) processor and 256 GB of main memory. All the algorithms were implemented in C# and run using Mono 4.2.3. We compared the following algorithms:

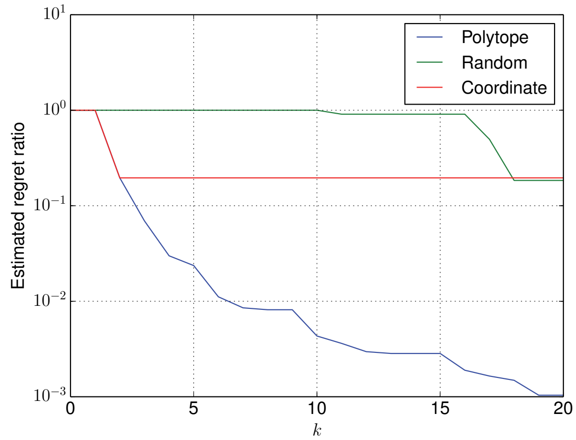


Figure 2: Regret ratio on movie summarization

- Coordinate (Algorithm 1).
- Polytope (Algorithm 2).
- Random: Given an approximation algorithm $\mathcal{A}$ for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$, we pick k random directions $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k$ and output the family $\{X_1, \ldots, X_k\}$ of solutions, where X_i is the output of $\mathcal{A}$ for $f_{\boldsymbol{a}_i}$.

We conducted experiments on a personalized recommendation problem and the budget allocation problem, which are described below in more detail. To estimate the regret ratio, we use Lemma 3, i.e., for each nonnegative normal vector $\boldsymbol{a}$, we compute an approximate solution for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$ and use it to compute an approximation to $\mathrm{rr}_{f_{\boldsymbol{a}}, \mathcal{C}}(\mathcal{S})$, and then we take their maximum.

Movie summarization

Let E be a set of movies, and suppose that we want to summarize them by making several lists of representative movies. When making such lists, we need to consider two conflicting criteria, that is, coverage and diversity. Here, the coverage of a list $S \subseteq E$ of movies is regarded as high if, for any movie i, there exist movies in S that are similar to i. The diversity of a list S is regarded as high if no two movies in S are similar. To define similarity between movies, we exploit users' ratings on the movies. That is, we represent each movie i by a vector $\boldsymbol{v}_i$ consisting of the users' ratings. Then, we measure the similarity $s_{i,j}$ between movies i and j by the inner product $\boldsymbol{v}_i^\top \boldsymbol{v}_j$. (Mirzasoleiman et al. 2016) proposed the following function $f : 2^E \to \mathbb{R}_+$ for measuring the quality of a list of movies:

$$f(S) = \sum_{i \in E} \sum_{j \in S} s_{i,j} - \lambda \sum_{i \in S} \sum_{j \in S} s_{i,j},$$

where $0 \leq \lambda \leq 1$. Here, the first and second terms represent the coverage and diversity, respectively, of S. Note that, if $s_{i,j}$'s are $\{0, 1\}$-valued and $\lambda = 1$, then the function f corresponds to a cut function.

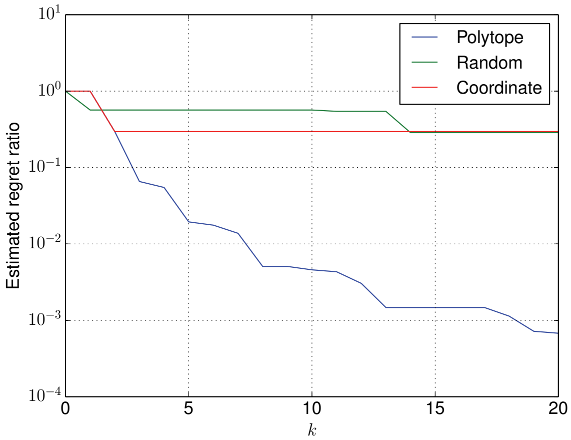


Figure 3: Regret ratio on budget allocation

Instead of maximizing f and outputting a single list, we consider computing a family of lists with a small regret ratio. To this end, we separate f into two objective functions $f_1, f_2 : 2^E \to \mathbb{R}_+$ defined as follows:

$$f_1(S) = \sum_{i \in E} \sum_{j \in S} s_{i,j},$$
$$f_2(S) = \lambda \sum_{i \in E} \sum_{j \in E} s_{i,j} - \lambda \sum_{i \in S} \sum_{j \in S} s_{i,j}.$$

Here, the term $\lambda \sum_{i \in E} \sum_{j \in E} s_{i,j}$ is introduced into f_2 in order to make it nonnegative. It is easy to see that f_1 is monotone submodular and f_2 is nonmonotone submodular.

In our experiment, we used the MovieLens 100K dataset, consisting of 100,000 ratings from 943 users on 1,682 movies (Grouplens 1998) and set $\lambda = 0.1$ as indicated in (Mirzasoleiman et al. 2016). We did not impose any constraint, i.e., $\mathcal{C} = 2^E$. As f_2 is a non-monotone function, we adopted the double greedy method (Buchbinder et al. 2012) as an approximation algorithm for $\max_{X \in \mathcal{C}} f_{\boldsymbol{a}}(X)$, whose approximation ratio is $1/2$.

Figure 2 shows how the regret ratio on f_1 and f_2 decreases for each method by increasing the size k of the output family (Coordinate outputs a family of size at most 2 for any k). We can observe that the regret ratio of Random stagnates around 0.2, which means that it is not sufficient to get lists of representative movies. Theorem 5 implies that Coordinate gives a regret ratio at most $1 - (1/2)/2 \approx 0.75$. Although the empirical performance is better, the obtained regret ratio of around 0.2 is still high. The regret ratio of Polytope decreases drastically as k increases. Even when k is as small as 20, the regret ratio is already around 0.001, which shows the superiority of Polytope over other methods. Although Corollary 9 states that the regret ratio decreases as $O(1/k)$, the empirical performance is even better.

Budget allocation

The budget allocation problem (Alon, Gamzu, and Tennenholtz 2012) models a marketing process that allocates a given budget among media channels, such as TV, newspapers, and the Web, in order to maximize the impact on customers.

We created a synthetic instance of the budget allocation problem as in (Soma et al. 2014). The instance can be represented as a pair of a bipartite graph $(L, R; E)$. Here, L is a set of 500 vertices and R is a set of 5,000 vertices. We regard a vertex in L as an ad source, and a vertex in R as a person. We then fix the degrees of vertices in L such that their distribution obeys the power law with $\gamma := 2.5$; i.e., the fraction of ad sources with out-degree d is proportional to $d^{-\gamma}$. For a vertex $u \in L$ of the supposed degree d, we choose d vertices in R uniformly at random and add edges to them.

Now, we construct two submodular functions $f_1, f_2 : 2^L \to \mathbb{R}_+$. The function f_1 is defined as

$$f_1(S) = \sum_{v \in R} \left(1 - 0.99^{|\Gamma(v) \cap S|}\right),$$

where $\Gamma(v) \subseteq L$ is the set of vertices connected to v. Consider the situation in which, by exploiting an ad source $u \in L$, for each person $v \in R$ connected to u, we have a chance of influencing v with probability 0.01. The function f_1 can then be regarded as the expected number of people influenced by the chosen ad sources; hence, we want to maximize f_1. It is known that f_1 is monotone submodular (Soma et al. 2014). We define $f_2 : 2^L \to \mathbb{R}$ as

$$f_2(S) = |L| - |S|,$$

which is the number of remaining ad sources. We want to maximize f_2 because it represents the saved cost. Note that f_2 is a non-monotone linear (and hence submodular) function.

In our experiment, we did not impose any constraint, i.e., $\mathcal{C} = 2^L$. As f_2 is a non-monotone function, we adopted the double greedy method (Buchbinder et al. 2012) again for the approximation algorithm.

Figure 3 shows how the regret ratio decreases using each method. The performance of each method is similar to that for the movie summarization task. Again, we can observe the superiority of Polytope.

Conclusions

In this work, we presented the coordinate-wise maximum method and polytope method for minimizing regret ratio in multi-objective submodular function maximization. Both methods have provable guarantees and we also showed that the trade-off attained by the polytope method in the biobjective case cannot be improved significantly. Further, we confirmed that the polytope method outperforms other methods through experiments conducted on a movie summarization problem and the budget allocation problem. Providing a theoretical guarantee of our method in the multi-objective setting is an interesting future work.

Acknowledgments T. S. is supported by JSPS Grant-in-Aid for Research Activity Start-up. Y. Y. is supported by JSPS Grant-in-Aid for Young Scientists (B) (No. 26730009), MEXT Grant-in-Aid for Scientific Research on Innovative Areas (No. 24106003), and JST, ERATO, Kawarabayashi Large Graph Project.

References

Alon, N.; Gamzu, I.; and Tennenholtz, M. 2012. Optimizing budget allocation among channels and influencers. In *WWW*, 381–388.

Buchbinder, N.; Feldman, M.; Naor, J. S.; and Schwartz, R. 2012. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS*, 649–658.

Clarkson, K. L., and Shor, P. W. 1989. Applications of random sampling in computational geometry, II. *Discrete & Computational Geometry* 4(5):387–421.

Daskalakis, C.; Diakonikolas, I.; and Yannakakis, M. 2010. How good is the chord algorithm? In *SODA*, 978–991.

Grouplens. 1998. MovieLens 100k dataset. http://grouplens.org/datasets/movielens/100k/.

Ko, C. W.; Lee, J.; and Queyranne, M. 1995. An exact algorithm for maximum entropy sampling. *Operations Research* 684–691.

Krause, A., and Horvitz, E. 2008. A utility-theoretic approach to privacy and personalization. In *AAAI*, 1181–1188.

Krause, A.; McMahon, H. B.; Guestrin, C.; and Gupta, A. 2008. Robust submodular observation selection. *Journal of Machine Learning Research* 9:2761–2801.

Krause, A.; Roper, A.; and Golovin, D. 2011. Randomized sensing in adversarial environments. In *IJCAI*, 2133–2139.

Krause, A.; Singh, A. P.; and Guestrin, C. 2008. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research* 235–284.

Lin, H., and Bilmes, J. 2010. Multi-document summarization via budgeted maximization of submodular functions. 912–920. NAACL-HLT.

Lin, H., and Bilmes, J. 2011. A class of submodular functions for document summarization. In *ACL-HLT*, 510–520.

Lin, X.; Yuan, Y.; Zhang, Q.; and Zhang, Y. 2007. Selecting stars: The k most representative skyline operator. In *ICDE*, 86–95.

Mirzasoleiman, B.; CH, E.; Karbasi, A.; and EDU, Y. 2016. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, 1358–1367.

Nanongkai, D.; Sarma, A. D.; Lall, A.; Lipton, R. J.; and Xu, J. 2010. Regret-minimizing representative databases. *Proceedings of the VLDB Endowment* 3(1-2):1114–1124.

Peng, P., and Wong, R. C.-W. 2014. Geometry approach for k-regret query. In *ICDE*, 772–783.

Soma, T.; Kakimura, N.; Inaba, K.; and Kawarabayashi, K. 2014. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *ICML*, 351–359.

Tao, Y.; Ding, L.; Lin, X.; and Pei, J. 2009. Distance-based representative skyline. In *ICDE*, 892–903.

Xia, T.; Zhang, D.; and Tao, Y. 2008. On skylining with flexible dominance relation. In *ICDE*, 1397–1399.

Yiu, M. L., and Mamoulis, N. 2009. Multi-dimensional top-k dominating queries. *The VLDB Journal* 18(3):695–718.

Yue, Y., and Guestrin, C. 2011. Linear submodular bandits and their application to diversified retrieval. In *NIPS*, 2483–2491.